

МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”

Факультет компьютерных наук  
Кафедра информационных систем

Сервис подачи работ на конференцию  
Курсовой проект  
по дисциплине  
Технологии программирования

09.03.21 Информационные системы и технологии  
Информационные системы и сетевые технологии

Обучающийся \_\_\_\_\_ В.А. Ушаков, 3 курс, д/о  
Обучающийся \_\_\_\_\_ С. Ю. Воронцова, 3 курс, д/о  
Обучающийся \_\_\_\_\_ К. И. Малышева, 3 курс, д/о  
Руководитель \_\_\_\_\_ Тарасов В. С.

Воронеж 2020

## Оглавление

Введение .....	3
Используемые определения.....	4
1. Постановка задачи .....	6
2. Анализ предметной области .....	8
3. Обоснование выбора продуктовых воронок .....	21
4. IDEF0 диаграмма.....	25
5. Диаграмма прецедентов.....	26
6. Диаграмма классов.....	29
7. Диаграмма объектов .....	30
8. Диаграмма последовательностей.....	31
9. Диаграмма взаимодействия.....	32
10. Диаграмма состояний.....	33
11. Диаграмма активностей .....	34
12. Диаграмма развертывания.....	36
13. Обоснование архитектуры проекта .....	37
14. ER – диаграмма.....	39
15. Тестирование.....	40
16. Реализация.....	42
17. Заключение.....	49
18. Приложение.....	50
19. Список использованных источников.....	51

## **Введение**

В современном мире любому специалисту, желающему оставаться конкурентоспособным в своей области, необходимо постоянно развивать свои навыки и доказывать свои компетенции. Существует несколько способов развития специалиста: прохождение тренингов, дополнительное обучение, митапы с коллегами, участие в научных конференциях. Остановимся на последнем.

На данный момент существует два варианта подачи статьи на научные конференции:

Первый вариант – организация подач статей через личную передачу файла или отсылки его на личную почту организаторов. Он достаточно прост в организации и поддержке, однако это сопряжено с сложностями как для организаторов, так и для участников. Для проверяющих возникает проблема хранения и ведения учета поданных статей. Для пользователя же это дополнительные затраты времени по написанию писем для проверок статуса его работ, звонки.

Второй вариант – создание сервиса, занимающегося автоматизацией подачи и учета поданных статей. У данного варианта есть ряд преимуществ:

- Редактирование своих загруженных работ под дистанционным руководством
- Удобное отслеживание состояния работы
- Удобство учета поданных работ и выставление им статуса

По данным причинам онлайн-сервис – это простое решение, с возможностью долгосрочной поддержки, которое доступно для большинства пользователей сети Интернет.

### Используемые определения

Таблица 1. Используемые определения

Веб-сервис	идентифицируемая уникальным веб-адресом (URL-адресом) программная система со стандартизированными интерфейсами, а также HTML-документ сайта, отображаемый браузером пользователя
Гость	неавторизованный на портале человек, пользующийся ограниченным функционалом веб-сервиса.
Пользователь	авторизованный на портале человек, пользующийся функционалом веб-сервиса.
Участник	пользователь, решивший принять участие в Конференции.
Редактор	пользователь, имеющий обязанности приема, загрузки, редактирования статей.
Администратор	человек, имеющий доступ к расширенному функционалу веб-сервиса, имеющий знания о формате приема статей

Header	прием использования изображения, текста и навигационных элементов на главной странице и закреплённых вверху веб-страницы.
Хэширование паролей	особое преобразование любого объема информации, в результате которого получается некое отображение, образ, называемый хэшем (hash) — уникальная короткая символьная строка, которая присуща только этому массиву входящей информации.
Hard skills	профессиональные навыки, которым можно научить и которые можно измерить. Для обучения hard skills необходимо усвоить знания и инструкции, качество обучения можно проверить с помощью экзамена
Soft skills	универсальные компетенции, которые гораздо труднее измерить количественными показателями. Иногда их называют личными качествами, потому что они зависят от характера человека и приобретаются с личным опытом.
Фреймворк	это программные продукты, которые упрощают создание и поддержку технически сложных или нагруженных проектов. Фреймворк, как правило, содержит только базовые программные модули

## 1. Постановка задачи

Целью данного курсового проекта является разработка самостоятельного сервиса для локальной Научной Конференции, который позволит упростить процесс приема и проверки статей, позволяет показывать актуальные новости, сроки, расписания Конференции, просматривать и скачивать научные архивы/сборники прошлых лет, дает возможность ознакомиться с официальной контактной информацией.

Для достижения данных целей сервис должен отвечать следующим требованиям:

1. Сервис должен иметь простой и понятный, неперегруженный динамическими элементами дизайн, выполненный в неброских цветах.
2. Сервис должен выполнять ряд основных функциональных задач:
  - 2.1. Создание средств для предоставления информации о конференции, ее новостях и сборниках научных работ прошлых лет, а также реализация средств добавления новостей и новых сборников администратором
  - 2.2. Реализация формы для создания статьи, реализация добавления информации с нее в базу данных, реализация возможности смены прикрепленного к статье файла, смены статуса статьи и его отображения
  - 2.3. Создание средств взаимодействия пользователей с организаторами конференции
  - 2.4. Возможность совершить Регистрацию/Авторизацию в системе
  - 2.5. Реализация всех ролей системы

Для выполнения данных требований необходимо выполнить следующие задачи:

1. Проектирование веб-сервиса средствами языка UML
2. Разработка back-end части, включающая в себя:
  - 2.1. Реализация ролей:
    - Администратор (Главный Организатор)
    - Редактор
    - Зарегистрированный пользователь (участник конференции)
    - Незарегистрированный пользователь (гость)
  - 2.2. Реализация функциональных возможностей ролей
  - 2.3. Подключение внешнего модуля для хранения данных
  - 2.4. Разработка базы данных
  - 2.5. Разработка функциональности статических и динамических страниц

3. Разработка front-end части, включающей в себя:

3.1. Создание макета дизайна

3.2. Реализация макета дизайна

4. Проведение тестирования проекта

## 2. Анализ предметной области

### 2.1. Анализ существующих решений

Анализ существующих решений будем проводить на основе данных сервисов:

Таблица 2. Примеры существующих решений

<a href="http://www.jinr.ru/about/events-plan/">http://www.jinr.ru/about/events-plan/</a>	Объединенный Институт Ядерных Исследований
<a href="http://econ.vsu.ru/conf/">http://econ.vsu.ru/conf/</a>	Конференция ВГУ по Экономике
<a href="https://indico.cern.ch">https://indico.cern.ch</a>	Индиго
<a href="https://lomonosov-msu.ru">https://lomonosov-msu.ru</a>	Сервис по созданию конференций от МГУ
<a href="https://tilda.cc/ru/tpls/sait-meropriyatiya/">https://tilda.cc/ru/tpls/sait-meropriyatiya/</a>	шаблон от Тильды

1. Сайт Объединенного Института Ядерных Исследований (<http://www.jinr.ru/about/events-plan/>)

Общее описание:

Данный сервис представляет собой сайт, предоставляющий информацию о деятельности ОИЯИ — международной межправительственной научно-исследовательской организации. Сайт ОИЯИ, главная страница которого представлена на Рисунок 1, выступает как полу-самостоятельный продукт, хранящий общую информацию на официальном сайте и проводящий регистрацию на сайте посреднике.



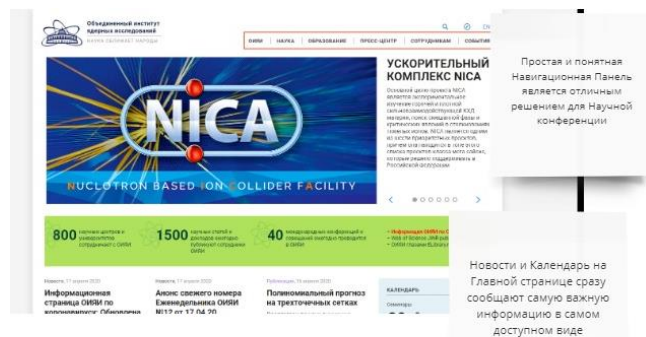


Рисунок 1.1 - Главная страница сайта ОИЯИ

На странице тематики исследований, представленной на Рисунок 1.2, присутствует раздел с архивом прошлых работ. Это полезный инструмент для всех заинтересовавшихся тематикой Конференции.

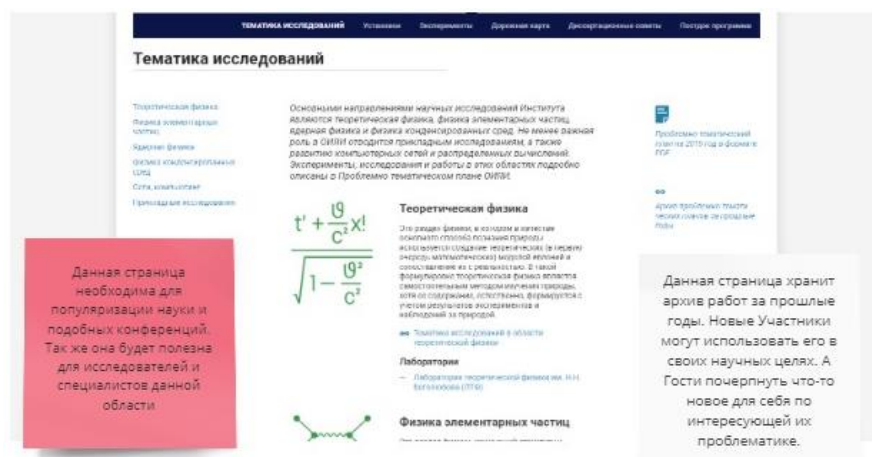


Рисунок 1.2 – Страница тематики исследований

#### Достоинства:

- понятный дизайн, отсутствие перенасыщенного дизайна (отсутствие динамичных элементов, анимаций)
- многостраничность
- разбиение на информационные блоки
- возможность просматривать старые работы и ранние сроки проведения Конференции

Недостатки:

- большая информационная загруженность - слишком много текстового контента
- не является самостоятельным ресурсом: нельзя произвести регистрацию на Конференцию
- использует сторонние ресурсы для регистрации участников

## 2. Сайт Научной Конференции по Экономике ВГУ (<http://econ.vsu.ru/conf/>):

Ежегодно на базе ВГУ проходит научная конференция по Экономике. Несмотря на достаточно устаревший дизайн, сайт конференции обладает всеми необходимыми функциями для подачи Докладов

Страница объявлений Рисунок выглядит простовато, но она хранит всю необходимую информацию, не перегружена лишними данными



Дизайн простой, но устаревший для 2020 года. Доступные и открытые функциональные возможности упрощают работу с сайтом. Статистика является интересным атрибутом, необходимым для анализа мероприятия

Данный формат больше подходит под наши нужды, т.к. Конференция является узконаправленной и по одному профилю.

Рисунок 2.1 – Страница новостей и объявлений

На странице регистрации участника, функциональные возможности точно соответствуют своим целям. Нет ничего лишнего и перегружающего страницу.

Интересная разработка на странице тезисов докладов, 2.2: страница содержит

**Воронежский государственный университет • ЭКОНОМИЧЕСКИЙ ФАКУЛЬТЕТ**  
**Студенческая конференция ЭФ ВГУ**

**Тезисы докладов**

Пожалуйста, ознакомьтесь с [требованиями к оформлению докладов](#). Файлы принимаются только в формате RTF, размер не более 256 Кбайт. Чтобы исправить доклад, удалите его и отправьте заново.

**Выберите секцию:**

- секция банков и банковской деятельности (закрыта)
- секция финансовых рынков и финансов организаций (закрыта)
- секция налогов и налогообложения (закрыта)
- секция экономики труда и социально-трудовой сферы (закрыта)
- секция проблем социально-экономического развития (для студ. неэкон. спец.) (закрыта)
- секция макроэкономических проблем современной экономики (закрыта)
- секция микроэкономических проблем современной экономики (закрыта)
- секция институциональных проблем современной экономики (закрыта)
- секция проблем развития предпринимательства и электронной коммерции (закрыта)
- секция управления персоналом (закрыта)
- секция бухгалтерского учета (закрыта)
- секция маркетинга (закрыта)
- секция проблем управления организациями (закрыта)
- секция экономики предприятий и предпринимательской деятельности (закрыта)
- секция информационных технологий и математических методов в экономике (закрыта)
- секция общего и стратегического менеджмента (закрыта)
- секция региональной экономики и территориального управления (закрыта)
- секция экономического анализа и аудита (закрыта)
- секция проблем бизнес-администрирования (для ВВА и MBA) (закрыта)
- секция экономической безопасности (закрыта)
- секция проблем корпоративного учета и корпоративной отчетности (закрыта)

Прикрепить файл (RTF ≤ 256 Кбайт):  Файл не выбран

**Отправленные доклады:**

Авторы и название доклада	Дата	Статус
<b>Секция проблем социально-экономического развития (для студ. неэкон. спец.)</b>		
1. Ушаков В.А., бака. 3 к. <a href="#">Автоматизированная система учета и обработки информации как основа повышения эффективности планирования экономики</a> Научный руководитель - к.э.н., доц., Павлова Е.А.	03.04. 23.20	оформл. ? содерж. ? принят ?
		удалить

Выделение красным цветом - файл заблокирован системой из-за отсутствия анкеты.  
 Возможно, неправильно указаны сведения об авторах в файле тезисов (см. [помощь](#)).  
 Условные обозначения в графе "Статус":  
 "?" - ожидает проверки,  
 "-" - не соответствует требованиям,  
 "+" - соответствует/принимается.

Как Участник, мы можем видеть наши доклады, их статусы и даты обновления. Но, было бы удобнее обозначать статус статьи цветом, а не графическим маркерам

Рисунок 2.2 - Страница тезисов докладов

информацию о статусе Доклада. Но, представлять статус в виде символов – не самый наглядный вариант. Лучшая замена в данном случае – цветовое представление. Например, «светофор», где красный цвет означает, что работа не принята, желтый – находится в разработке, зеленый – принята.



- устаревший дизайн
- отсутствует возможность обсуждения работы

### 3. Indico (<https://indico.cern.ch>)

Сервис помогает создавать и управлять конференциями, встречами и лекциями. Может являться сайтом-посредником, как, например, для конференции Института Ядерных Технологий.

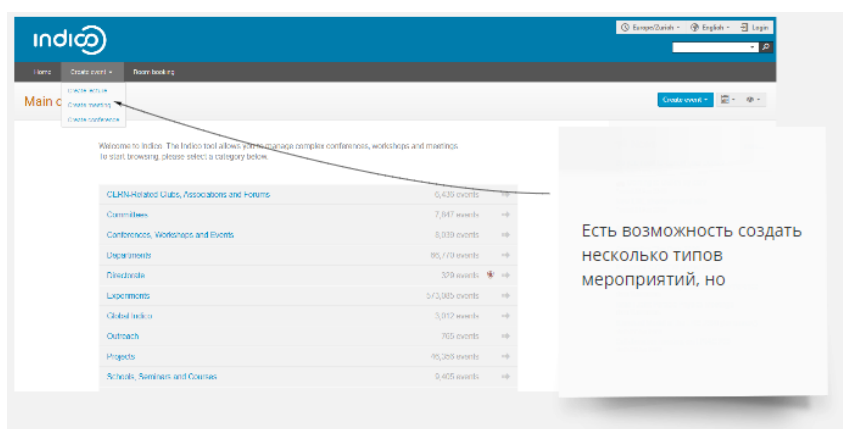


Рисунок 3.1 - Главная страница [indico.cern.ch](https://indico.cern.ch)

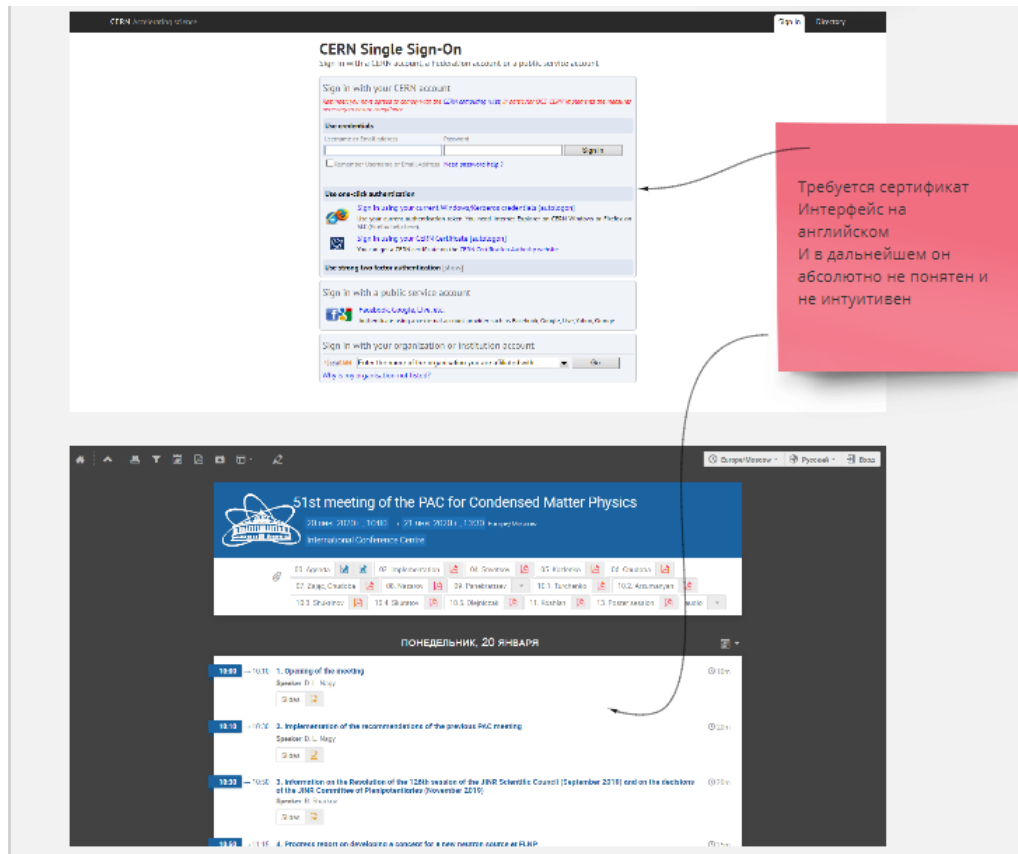


Рисунок 3.2 - Страницы сайта indico.cern.ch

#### Достоинства:

- возможность создать несколько типов мероприятий

#### Недостатки:

- непонятный и сложный графический интерфейс
- необходимо получение сертификата
- использование иностранного языка усложняет ориентирование на сайте

4. Сервис от МГУ на основе Олимпиады имени Ломоносова (<https://lomonosov-msu.ru>):

Бесплатный сервис по организации мероприятий. Максимально приближенный к нашему проекту. Но при создании своего мероприятия необходимо пройти модерации. К тому же, дизайн мероприятия единый для всех, поэтому наша Конференция не будет отличаться от других. Достоинствами данного сервиса являются его функциональные возможности.

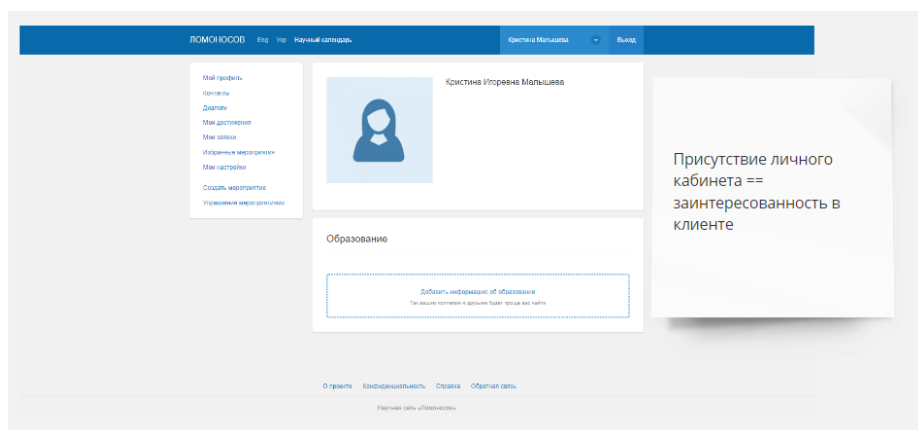


Рисунок 4.1 - Личный кабинет пользователя

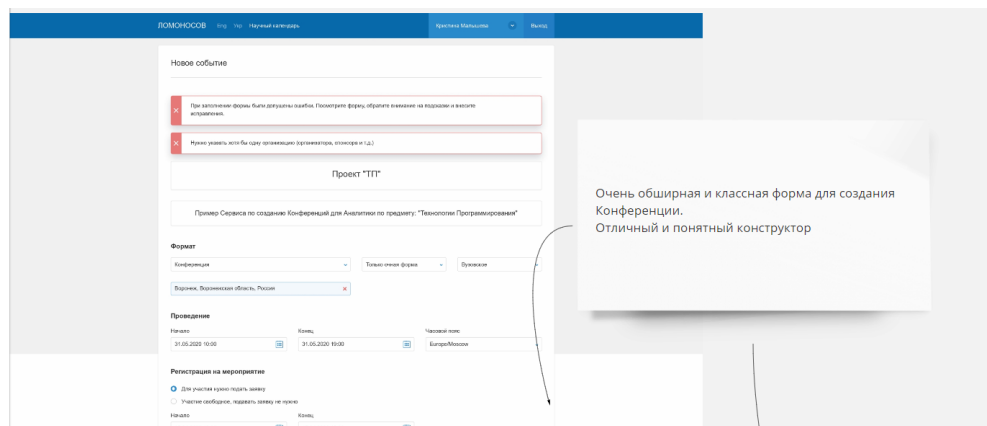


Рисунок 4.2.1 - Анкета создания мероприятия

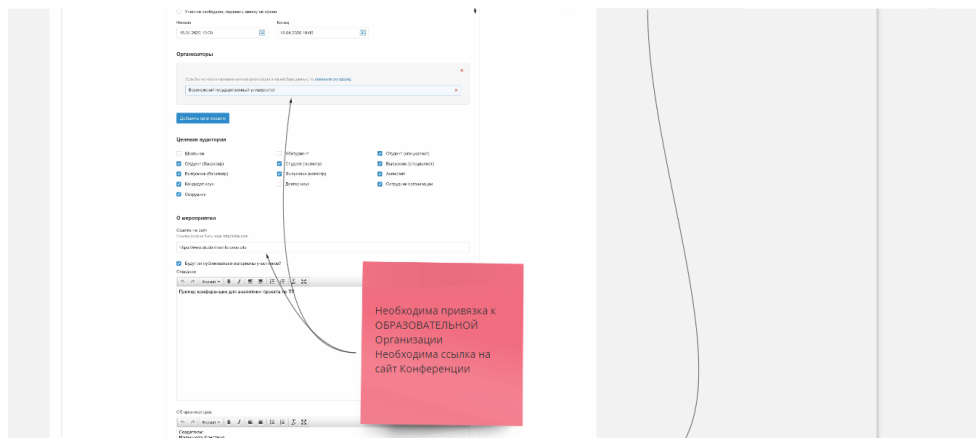


Рисунок 4.2.2- Анкета создания мероприятия (продолжение)

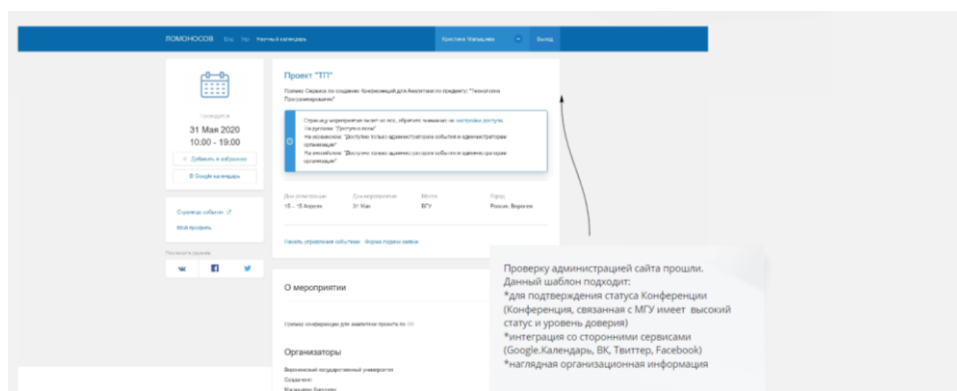


Рисунок 4.3 - Страница готового мероприятия

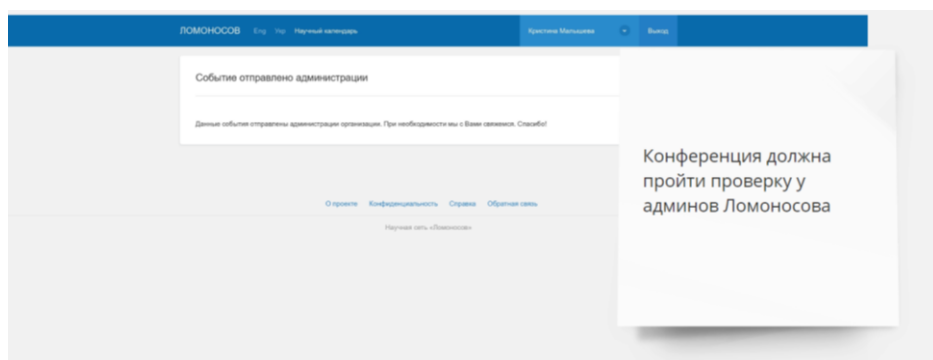


Рисунок 4.4 - Оповещение о передаче данных администратору



#### Достоинства:

- наличие личного кабинета
- возможность создавать Конференцию как конструктор
- статус
- интеграция со сторонними сервисами
- можно собирать и скачивать заявки и доклады
- есть чат общения с участниками

#### Недостатки:

- привязка к образовательной организации
- неизвестно, пройдет ли наша Конференция модерации
- неотличительный дизайн

#### 5. Tilda Publishing – как пример конструктора сайтов (<https://tilda.cc/ru/tpls/sait-meropriyatiya/>):

Этот конструктор делает акцент на типографике и возможностях детальной настройки доступных блоков, которых всего около 450. Функциональные возможности так же широки, но недостаточны.

Тильда предлагает использовать макет, подобный Рисунок 5.1 - Одностраничный сервис по созданию мероприятия как можно видеть данный макет предоставляет всю

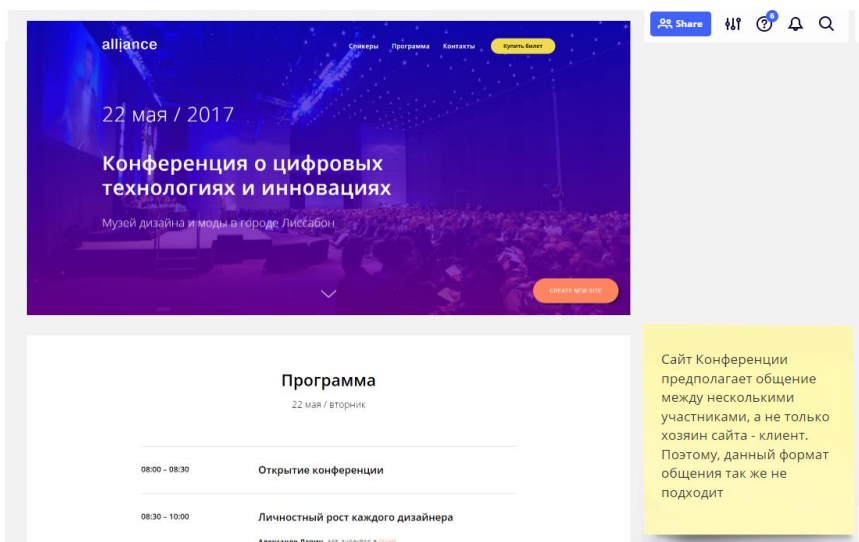


Рисунок 5.1 - Одностраничный сервис по созданию мероприятия

Прошлогодняя конференция

Фотографии с нашей бизнес-конференции, которая прошла в 2016 году

Создать новый пост

The diagram illustrates a central system for data collection and management, represented by a large yellow rectangle. This central system is connected to several external services, each represented by a smaller yellow rectangle. The connections are as follows:

- Forms of data collection and management of client data:** This section is divided into three sub-sections:
  - Constructor forms:** "Создает свои формы сбора данных. Собирает email, телефон, количество заказов и многое другое. Можно дать форму для клиентов сайта, подключить систему мониторинга."
  - Instantaneous notifications in Telegram or Slack:** "Получает заявки и мгновенно сообщает в мессенджер, в каком статусе заказ, кто позвонил клиенту и какие действия."
  - Forming a list of email recipients:** "Учитывает форму в Telegram и собирает список клиентов в таблицу Excel, Outlook, Mailchimp, Битрикс24."
- Obtaining information from the customer directly on the site:** "Устанавливает в сервисе формы для email и телефон, автоматически собирает данные, формирует список."
- Convenient and fast integration with existing services, such as Google.Disk and simple management of the database:** "Бд очень простая, но абсолютно не подходит для серьезного проекта, рассчитанного не на один год использования."
- Placing orders, using AmoCRM, Megaplan, Bitrix24:** "Создает интеграцию данных клиентов (CRM) и формирует в Telegram "Просто поставьте ваш email"."

Но уточнив требования Тильды можно заключить что ее возможностей не хватает для реализации задуманного нами проекта.

Достоинства:

- возможность создавать Конференцию как конструктор
- яркий дизайн

Рисунок 5.3 - Страница Формы сбора данных и управления базой данных клиента

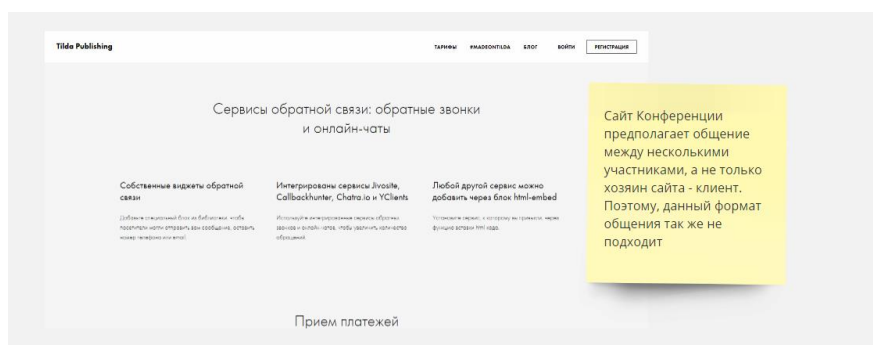


Рисунок 5.4 - Страница Сервиса обратной связи

- Недостатки:
- недостаточный функциональные возможности
  - подходит для создания сайтов-визиток, но не серьезного приложения

Итог на основе анализа данных сервисов:

Каких аспектов следует придерживаться при создании Сервиса:

- понятный дизайн, отсутствие перенасыщенного дизайна (отсутствие динамичных элементов, анимации, дизайн не превалирует над функц. частью)
- многостраничность
- разбиение на информационные блоки
- наличие личного кабинета
- абсолютно самостоятельный ресурс
- должна присутствовать возможность присылать свои Доклады
- доступные и открытые функциональные возможности
- есть возможность просматривать статус докладов

- присутствие аналитики Конференции
- простая интеграция с другими сервисами
- возможность просматривать старые работы и ранние сроки проведения Конференции
- чат для общения
- возможность собирать и скачивать доклады

Каких аспектов следует избегать:

- недостаточные функциональные возможности
- непонятный и недружелюбный графический интерфейс
- необходимо получение сертификата
- устаревший дизайн
- отсутствует возможность обсуждения поданной работы
- привязка к какой-либо организации
- большая информационная загруженность, отсутствие разбиения на информационные блоки
- не является самостоятельным ресурсом: нельзя произвести регистрацию на конференцию
- использовать цветовые элементы в качестве регистров состояний

## 2.2. Анализ потребности

На анализе рынка можно заключить, что общих альтернатив нет. Существуют только индивидуальные. Т.к. цель курсового проекта– создание независимого продукта, мы не желаем использовать сайт-посредники, размещать данные об Участниках на сторонних ресурсах.

К тому же, в будущем, с ростом численности участников и гостей Конференции возможно расширение функций Сервиса, что, при использовании сайтов-посредников и конструкторов может привести к перегрузкам сервера, а, следовательно, и к падению мощностей Сервиса.

### 3. Обоснование выбора продуктовых воронок

С помощью сервиса Яндекс. Метрика было принято решение создать три продуктовых воронки, основанных на целевых действиях пользователя Сервиса.

Также принимая во внимание 2. Анализ предметной области и личный интерес разработчиков Сервиса, выделены три сценария пользователя, ссылки на которые представлены в Приложении:

1. Ознакомление с общедоступной информацией
2. Отправка сообщений
3. Подача статей Участниками Конференции

Причины выбора сценариев:

#### 1. Общие причины:

- Улучшение графического интерфейса
- Улучшение функциональных возможностей (удаление неиспользованных функциональных элементов, если такие обнаружены и добавление новых и/или необходимых элементов, и/или обновление существующих)
- Понимание поведения среднестатистического зарегистрированного/незарегистрированного пользователя Сервиса (как люди попадают на сайт Сервиса, что это за люди (какой возраст, какие устройства) и как они работают со страницами внутри сайта)

Добавлено примечание ([U1]): Изменить

#### 2. Частные причины:

Таблица 3. Частные причины выбора данных воронок

Ознакомление с общедоступной информацией	1. Анализ количества Гостей 2. Анализ кол-ва Гостей, проявивших интерес к участию в Конференции, путем ознакомления с общедоступной информацией
--	--

Отправка сообщений	1. Анализ данной последовательности действий необходим для анализа поведения пользователя при использовании чата в целях взаимодействия с организаторами конференции
Подача статей Участниками Конференции	1. Подсчет кол-ва отправлений формы подачи статьи, путем нажатия пользователем кнопки «Отправить» 2. Определение сложности подачи статьи через данную форму

#### Результаты работы воронок:



Рисунок 6 - Яндекс.Метрика. Ознакомление с общедоступной информацией

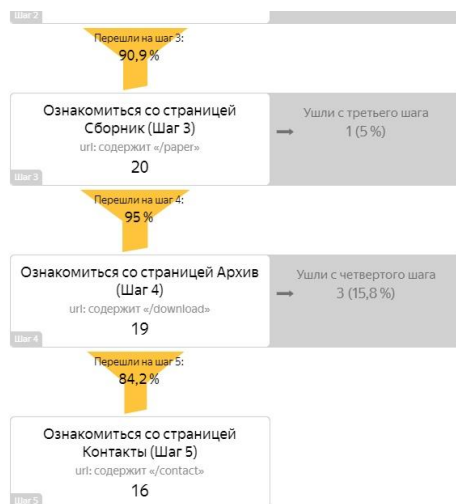


Рисунок 7 - Яндекс.Метрика. Ознакомление с общедоступной информацией. Продолжение

Вывод об анализе воронки «Ознакомление с общедоступной информацией»:

Большинство пользователей заинтересовано в участии в Конференции, Благодаря удобному и интуитивно понятному расположению элементов навигации и управления.

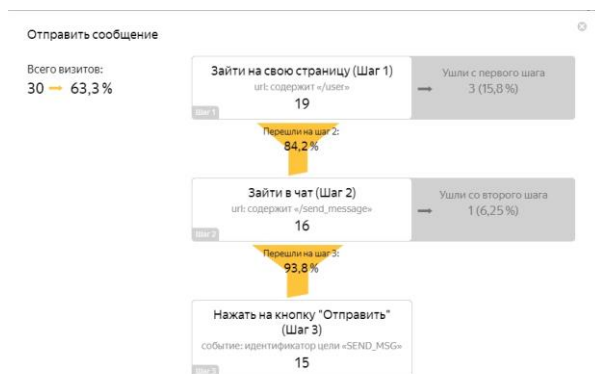


Рисунок 8 - Яндекс.Метрика. Отправить сообщение

Вывод об анализе воронки «Отправить сообщение»: 2/3 зарегистрированных пользователей смогли разобраться с процессом подачи статей на Конференцию и 7/9

пользователей перешли в чат, чуть меньше отправили сообщения. Функция действительно пользуется спросом среди Участников.

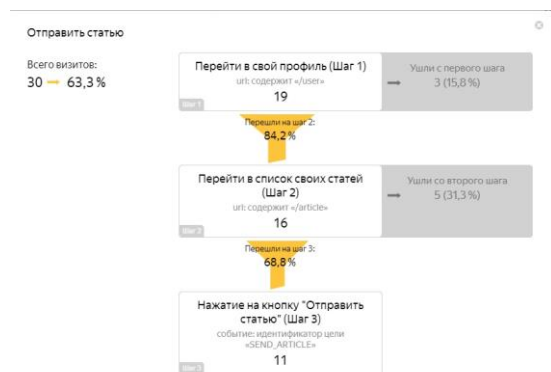


Рисунок 9 - Яндекс. Метрика. Отправить статью

Вывод об анализе воронки «Отправить статью»: 11/19 пользователей успешно справились с отправкой формы. Однако, около 30% пользователей не справились с данной операцией, что может быть связано как с незаинтересованностью пользователей в данной операции или с непониманием интерфейса страницы.



#### 4. IDEF0 диаграмма

Рассмотрим основной бизнес – процесс на примере контекстной диаграммы, представленной на Рисунок 10 - Контекстная диаграмма. Данная диаграмма представляет собой общее видение процесса работы Сервиса.

Работу сервиса регулируют Правила Конференции и Пользовательское соглашение, которые определяются совместно с Организатором Конференции, Законы РФ, Уровни доступа к Системе, которые определяются Администратором сервиса.

Обеспечивают работу Системы Администратор Сервиса, Организатор Конференции и Редакторы (для Конференции должен быть заявлен минимум один Редактор) и Исходная БД.

На вход в Систему поступает Докладчик, причем, Докладчиком выступает незарегистрированный пользователь, который в дальнейшем может принять одну из трех ролей: Администратор, Редактор, Участник. Т.е. не только Участник может выступать на Конференции, но и Администратор и Редактор. На выходе Система выдает Утвержденные/Неутвержденные доклады.

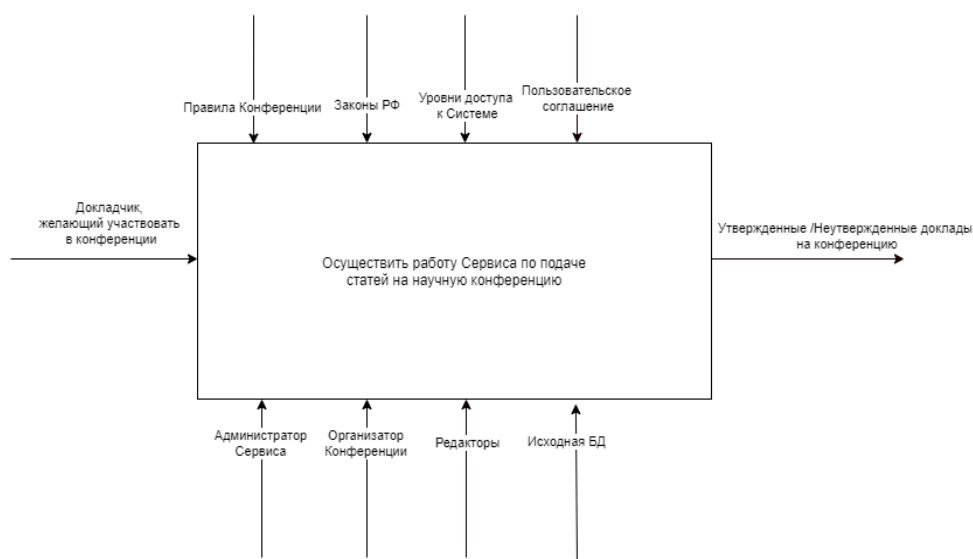


Рисунок 10 - Контекстная диаграмма

## 5. Диаграмма прецедентов

Диаграммы прецедентов показывают действия и отношения актеров, т.е. действующих лиц системы, между собой и их действия по отношению к системе.

На Рисунок 11 – представлены действующие лица со связями наследования. Как можно видеть, самым большим набором функций обладает Зарегистрированный, а самым малым – Незарегистрированный пользователь. Это означает, что функции Незарегистрированного пользователя наследует Зарегистрированный.

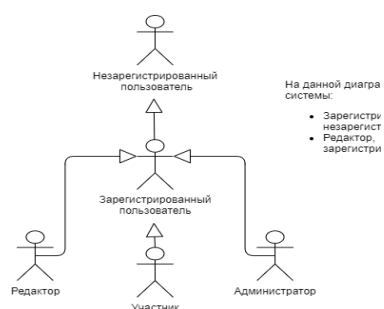


Рисунок 11 – Иерархия пользователей

На рисунках 12, 13 и 14 представлены действующие лица со связями наследования, а также их основные функции:

### Функции незарегистрированного пользователя (Гостя)

- Гостю доступен просмотр вводной информации (страниц Архив, Контакты, Главное)
- Гостю доступен просмотр сборников
- Гостю доступно скачивание сборников
- Гостю доступен просмотр Новостей
- Гость способен зарегистрироваться

### Функции зарегистрированного пользователя (участника)

- Наследует функциональные возможности Гостя
- Пользователю может совершить вход/выход
- Отправлять сообщения Администратору
- Отправлять Сообщения Редактору
- Загружать файлы формата .pdf, .docs, .rtf на веб-сервис
- Сменить файл, который прикреплен к статье

- Изменить свой профиль (изменить Пароль, Имя, Фамилия, о Себе)
- Заполнение формы для подачи статьи
- Смотреть статус своих статей (принята/не принята/в разработке)
- Просматривать статистику о своих работах

#### Функции зарегистрированного пользователя (редактора)

- Наследует функциональные возможности участника
- Отправить сообщение Участнику
- Просматривать файлы у статей участников
- Изменять статус статей участников (принята/не принята/в разработке)
- Просматривать статистику работ участников

#### Функции зарегистрированного пользователя (администратора)

- Наследует функциональные возможности редактора
- Добавляет новости на сайт
- Добавляет сборники
- Меняет роли пользователей
- Может реализовать 3 блокировки: запретить зарегистрированному пользователю присылать сообщения и/или загружать файлы и/или создавать статьи

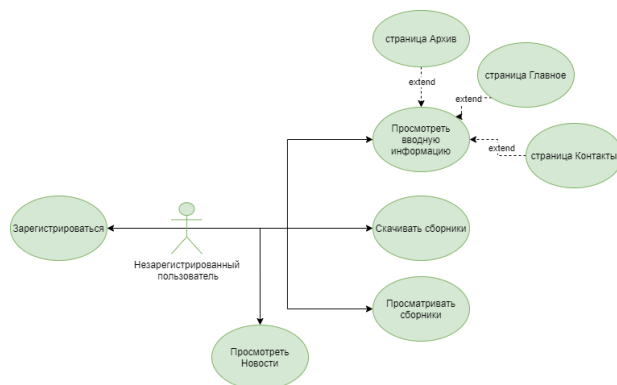


Рисунок 12 - Use-case Незарегистрированного пользователя

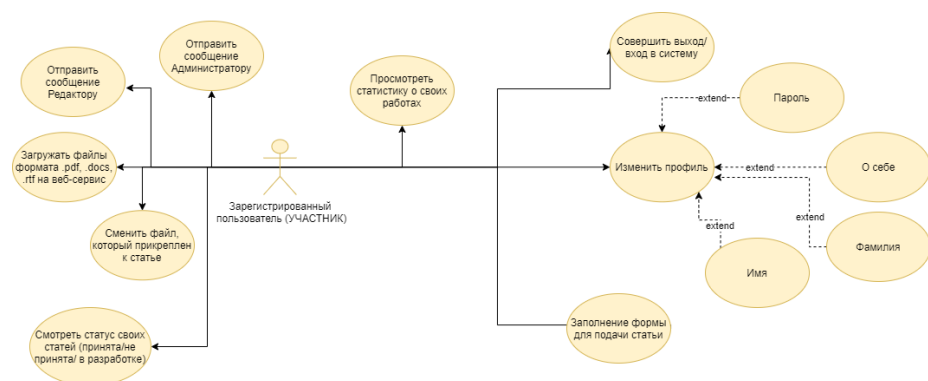


Рисунок 13 - Use-case Зарегистрированного пользователя (Участника)

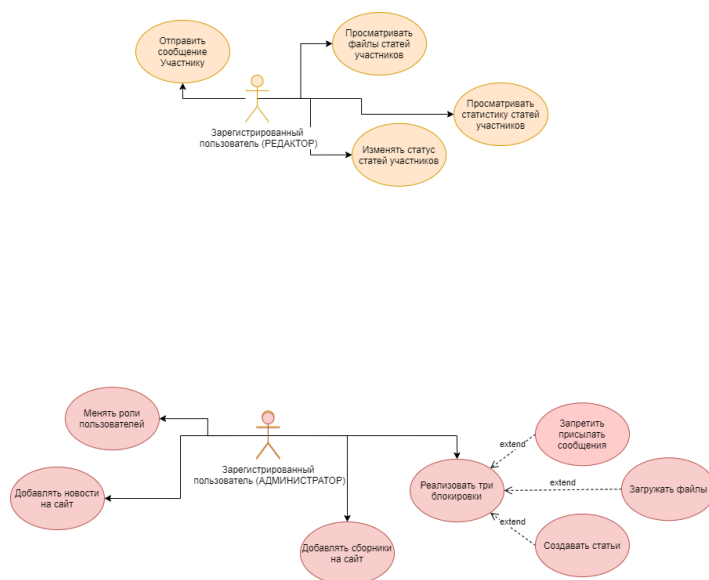


Рисунок 14 - Use-case Зарегистрированного пользователя (Редактора и Администратора)

## 6. Диаграмма классов

Диаграмма классов показывает классы системы, их атрибуты, методы и взаимодействия. Она представлена на Рисунк 15 - Диаграмма классов.

В качестве примера детальнее рассмотрим класс «BlockUser». Данный класс используется при обеспечении работы функций Администратора.

Класс имеет атрибуты: и тип данных:

- id: integer
- id: user
- block\_message: boolean, block\_article: boolean, block\_file: boolean
- И методы: и из возвращающий тип:
- add\_blockuser(user\_id, block\_message, block\_article, block\_file): void
- add\_blockuser(user\_id, block\_message, block\_article, block\_file): void
- check\_user(user\_id): bool
- +delete\_block(user\_id, block\_message, block\_article, block\_file): void

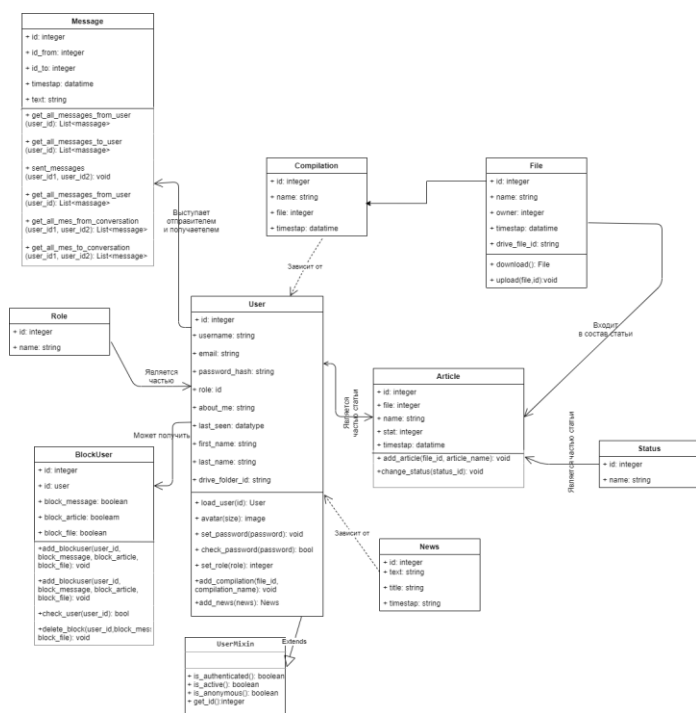


Рисунок 15 - Диаграмма классов

### 7. Диаграмма объектов

Диаграмма объектов на Рисунок 16 - Диаграмма объектов показывают множество объектов - экземпляров классов и отношений между ними в некоторый момент времени. Она поясняет и дополняет диаграмму классов.

Продолжим пояснение класса «BlockUser» на примере объекта «NewBlockRegisteredUser».

Он имеет уже конкретные атрибуты с конкретными значениями:

- id: 5
- id: 47
- block\_message: True
- block\_article: False
- block\_file: False

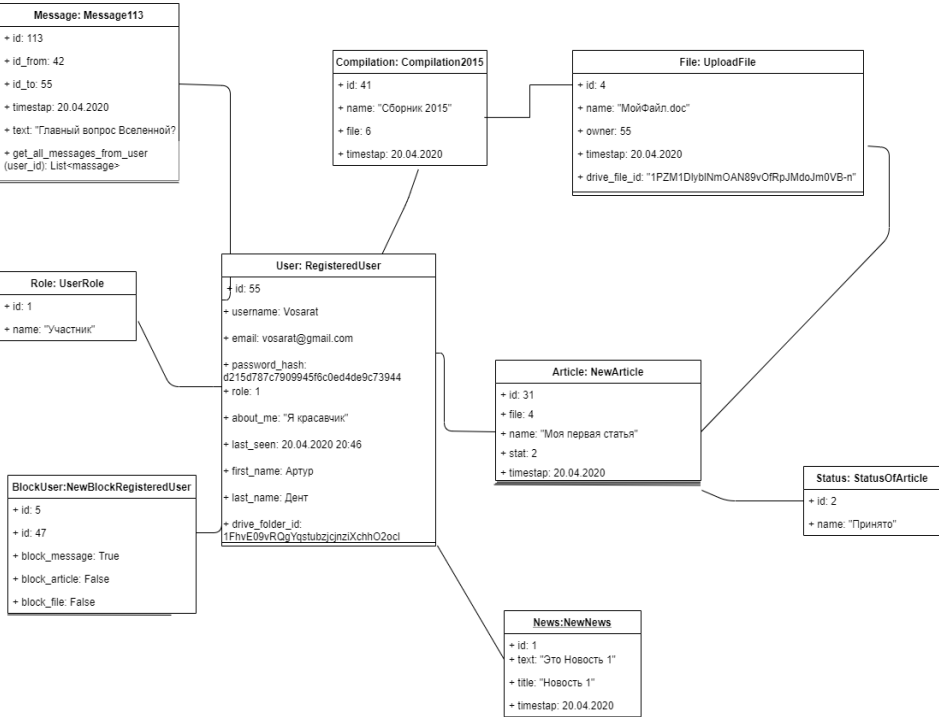


Рисунок 16 - Диаграмма объектов

## 8. Диаграмма последовательностей

Диаграмма последовательности отображает взаимодействие объектов в динамике. Т.е. диаграмма последовательностей отображает временные особенности передачи и приема сообщений объектами. На Рисунок 17 - Диаграмма последовательностей. Подача и проверка статьи представлен пример диаграммы для одной последовательности – подачи статьи.

Изначально Участник подает запрос Системе на добавление статьи, после приема запроса, Система отправляет уведомление Участнику. Затем Система создает форму для создания статьи, которую Участнику необходимо заполнить и к которой необходимо прикрепить файл (физическая статья). После передачи информации Системе, Форма самоуничтожается. Система, в свою очередь, отправляет статью хранится на Google.Disk, в Google.Disk обратно отправляет идентификатор файла. Если статью(или статьи) нужно проверить, редактор обращается к Системе и получает список последних, проверяет и отправляет в Систему уже Статус статьи. После всего Система оповещает Участника об изменённом статусе статьи. А Участник по желанию начинает вносить правки в статью.

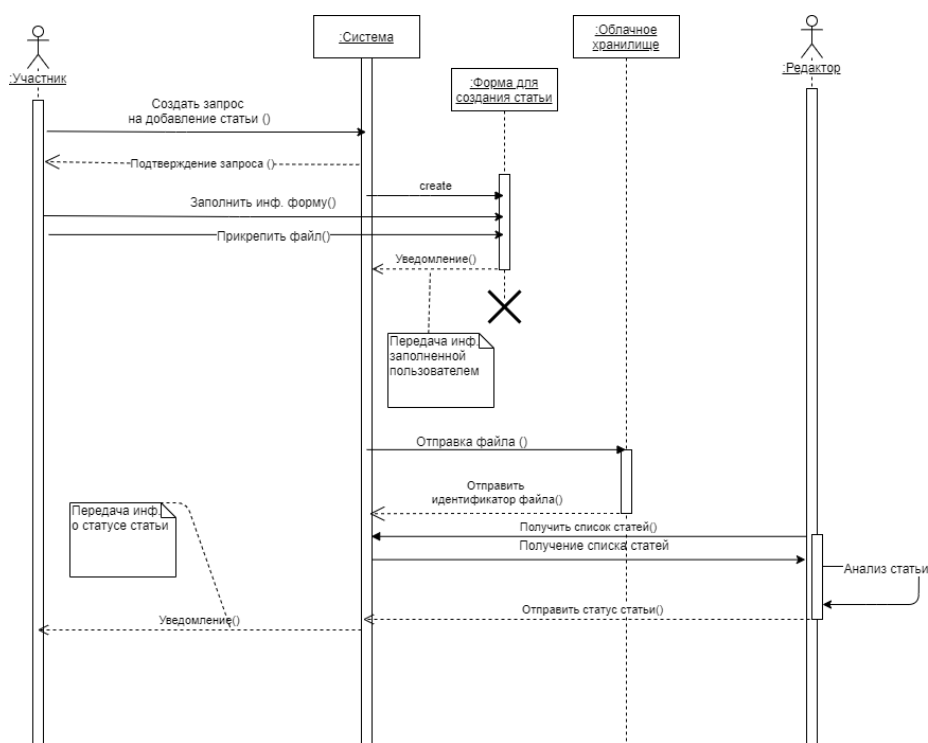


Рисунок 17 - Диаграмма последовательностей. Подача и проверка статьи

## 9. Диаграмма взаимодействия

Необходимо теперь определить, какие способы взаимодействия с системой есть в распоряжении пользователей. Она позволяет видеть все взаимодействия запросов в системе и дополняет Рисунок 18 - Диаграмма взаимодействия.

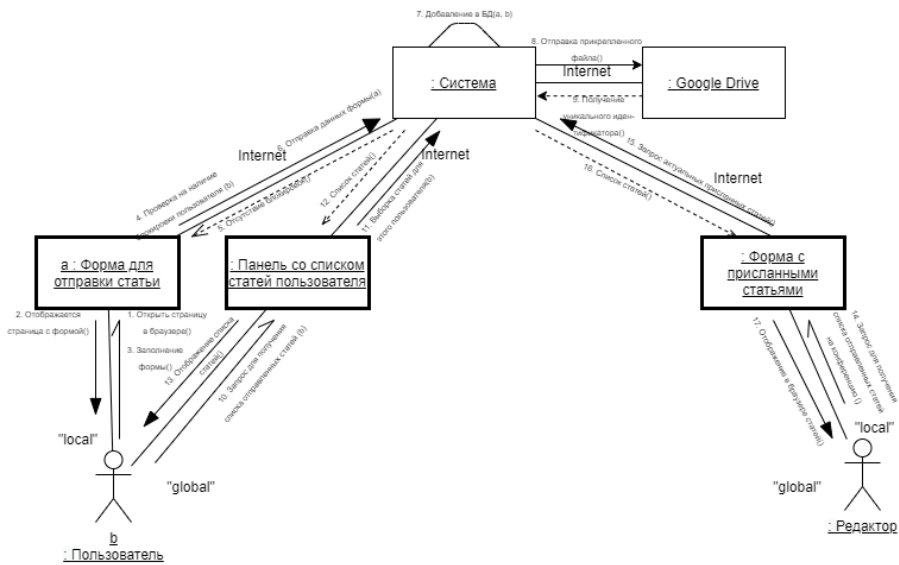


Рисунок 18 - Диаграмма взаимодействия



## 10. Диаграмма состояний

Диаграмма состояний показывает, как объект переходит из одного состояния в другое. На Рисунок 20 - Диаграмма состояний. Форма показано изменение состояний Формы, в то время как на Рисунок 19 - Диаграмма состояний. Статьяизменение состояний Статьи.

Рассмотрим Рисунок 19 - Диаграмма состояний. Статья

после состояния Проверяется, статья может принять одно из двух конечных состояний:

- Принята
- Не принята

Причем, каждому состоянию соответствует статус статьи. И если состоянию Принята в соответствие ставится конечный статус «Принята», состоянию Не принята - «Не принята», то статусу «В доработке» - ставится незавершенное, циклическое состояние – Дорабатывается. Т.е. в итоге, статья будет либо будет участвовать в Конференции, либо совсем нет, что соответствует Рисунок 10 - Контекстная диаграмма.

Описание состояния одного объекта: Формы, хранящей инф. о научной статье



Рисунок 20 - Диаграмма состояний. Форма

Описание состояния одного объекта: статьи

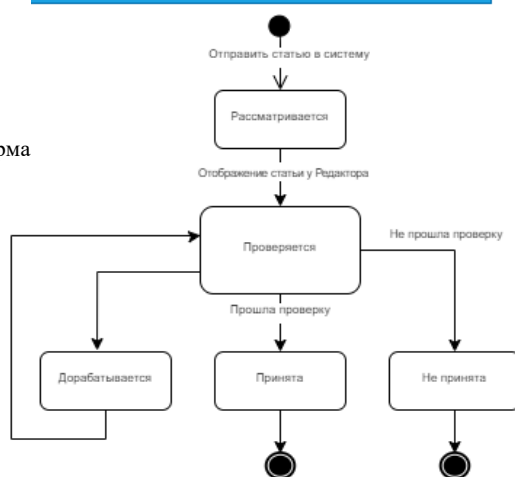


Рисунок 19 - Диаграмма состояний. Статья

11. Диаграмма активностей

Моделируя поведение проектируемой системы, часто недостаточно изобразить последовательности, а нужно также раскрыть детали алгоритмической реализации операций.

Диаграммы активностей для проверки статьи и подачи статьи, представленные на Рисунок 21 - Диаграмма активностей. Проверка статьи и Рисунок 22 - Диаграмма активности. Подача статьи соответственно поясняют Рисунок 17 - Диаграмма последовательностей. Подача и проверка статьи.

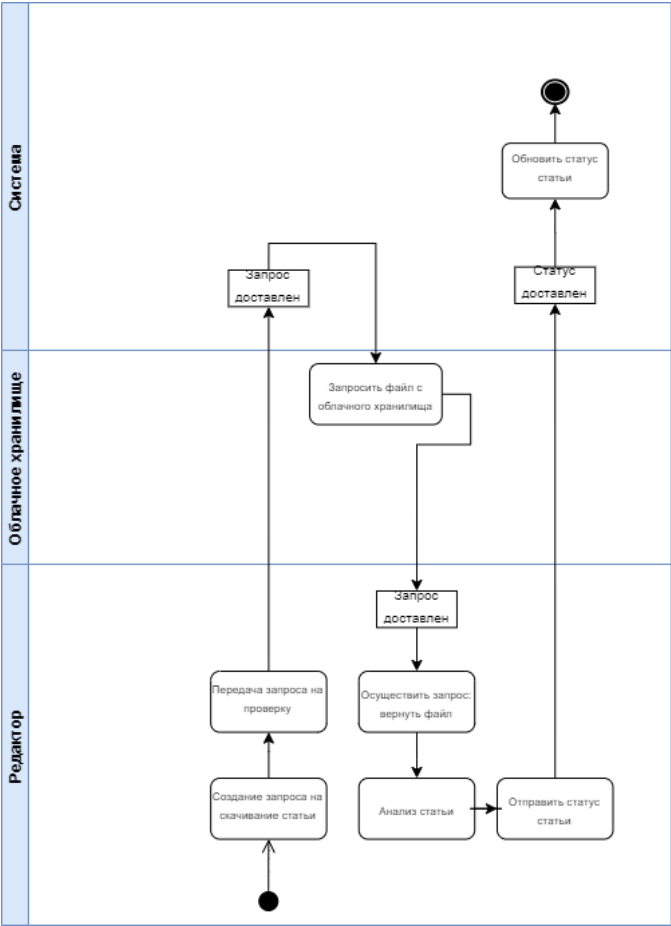


Рисунок 21 - Диаграмма активностей. Проверка статьи

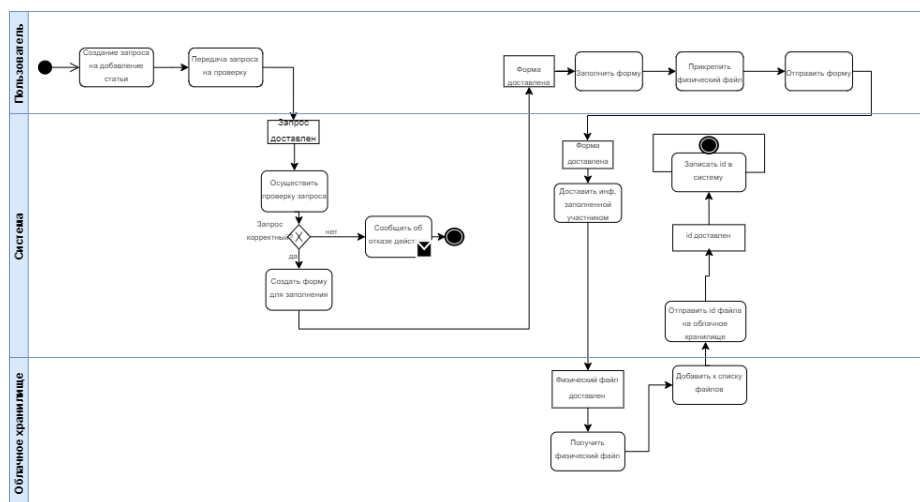


Рисунок 22 - Диаграмма активности. Подача статьи

## 12. Диаграмма развертывания

Диаграмма развертывания на Рисунок 23 - Диаграмма развертывания должна показывать, какие аппаратные компоненты существуют, какие программные компоненты работают на каждом узле, и как различные части этого комплекса соединяются друг с другом.

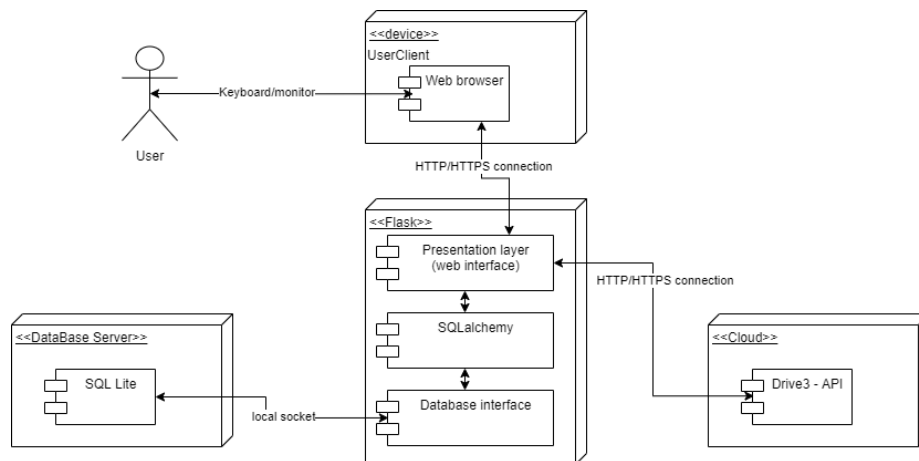


Рисунок 23 - Диаграмма развертывания

### 13. Обоснование архитектуры проекта

Архитектура проекта: при проектировании приложения был использован паттерн проектирования MVT (одна из модификаций MVC). ORM в проекте использует паттерн Unit of Work.

Паттерн MVT (Model-View-Template) является модификацией MVC. Основное различие между этими двумя шаблонами состоит в том, что Flask сам заботится о части контроллера (программный код, который контролирует взаимодействие между моделью и представлением). Шаблон представляет собой файл HTML, смешанный с языком шаблонов Jinja2. Схематично паттерн можно определить следующим образом:

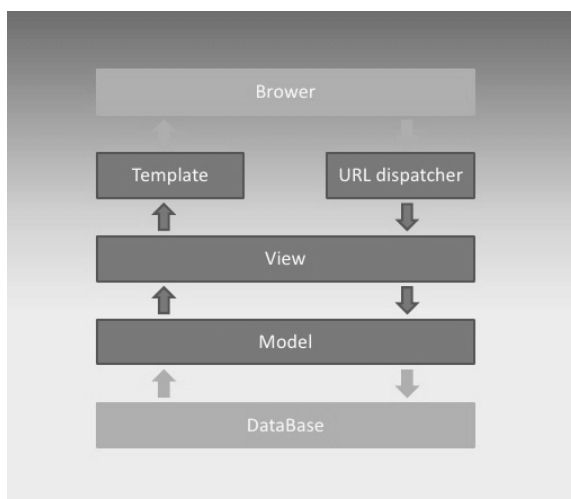


Рисунок 24 - Схема паттерна MVT

Основные элементы паттерна:

URL dispatcher: при получении запроса на основании запрошенного адреса URL определяет, какой ресурс должен обрабатывать данный запрос.

View - это интерфейс: получает запрос, обрабатывает его и отправляет в ответ пользователю некоторый ответ. Если для обработки запроса необходимо обращение к модели и базе данных, то View взаимодействует с ними. Для создания ответа может применять Template или шаблоны. В архитектуре MVC этому компоненту соответствуют контроллеры (но не представления).

Model: описывает данные, используемые в приложении. Отдельные классы, как правило, соответствуют таблицам в базе данных.

Template: представляет логику представления в виде сгенерированной разметки html. В MVC этому компоненту соответствует View, то есть представления.

Причины выбора данного паттерна:

- четко разделена логика приложения
- облегчается поддержка и тестирование кода
- возможность автоматизации части контроллера во Flask

Аналогами выступали паттерны Model-View-Presenter и Model-View-View Model.

Причины, почему не были выбраны данные паттерны:

- Отсутствие предрасположенности выбранного фреймворка под данные паттерны
- Неоправданная сложность для данного проекта

#### 14. ER – диаграмма

При известных процессах, объектах и ролях необходимо определить, какие данные нужно хранить в базе. Схема базы данных определяет таблицы, поля и ограничения целостности. Данная схема представлена на Рисунок 25 - ER-диаграмма приложения:

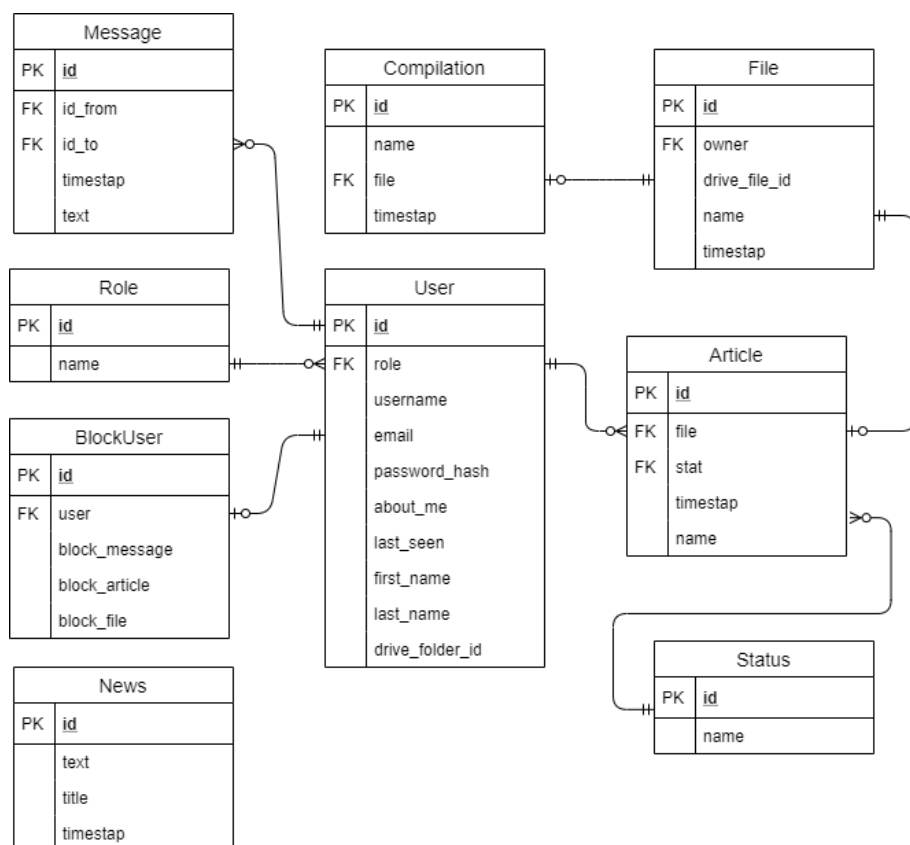


Рисунок 25 - ER-диаграмма приложения

## 15. Тестирование

Результаты тестирования представлены в отчете Тест-кейсы.

Использовалось функциональное тестирование.

Тестирование проводилось по основным рабочим сценариям.

Примеры тестирования и Общий отчет тестирования представлены ниже:

Актёры	Описание	% ТК выполнено	% ТК пройдено	Приоритет	Комментарии
Гость	ТК проходили только незарегистрированные пользователи	100	80	High	Все непройденные ТК исправлены и проходят исправно
Участник	ТК проходили только зарегистрированные пользователи	100	100	High	Все непройденные ТК исправлены и проходят исправно
Редактор	ТК проходили только зарегистрированные пользователи, обладающие правами редактора	100	50	High	Все непройденные ТК исправлены и проходят исправно
Администратор	ТК проходили только зарегистрированные пользователи, обладающие правами администратора	100	75	High	Все непройденные ТК исправлены и проходят исправно

Рисунок 27 - Общий отчет о тестировании

Идентификатор	g-01	<p><b>Предусловие:</b> Пользователь не авторизован Нажать на кнопку Регистрация на Навигационной Панели</p> <p><b>Шаги:</b> 1. Заполнить поле Логин валидным значением 2. Заполнить поле email валидным значением 3. Заполнить поле Имя валидным значением 4. Заполнить поле Фамилия валидным значением 5. Заполнить поле Пароль валидным значением 6. Заполнить поле Повторить пароль валидным значением, совпадающим со значением поля Пароль 7. Нажать на кнопку Регистрация</p> <p><b>Ожидаемый результат:</b> Открывается окно с приветственной информацией, сообщаемое об успешной регистрации</p> <p><b>Фактический результат:</b> Происходит падение после нажатия на кнопку Регистрация. Внутренняя ошибка 500</p>
Название	Падение при регистрации	
Приоритет	Высокий	
Устройство	Омега 68.0	
Версия приложения	26.05.2020	
Серьезность	Критическая	
Дополнения		

Рисунок 26 - Тестирование Регистрации

Идентификатор	uch-02	<p><b>Предусловие:</b> Пользователь авторизован Пользователь находится в Моем Кабинете Пользователь нажал на строку Ваши Статьи</p> <p><b>Шаги:</b> 1. Заполнить текстовое поле Название статьи валидным значением 2. Нажать на кнопку Выберите файл 3. Загрузить файл формата .doc 4. Нажать на кнопку Отправить</p> <p><b>Ожидаемый результат:</b> Происходит обновление страницы, появляется форма статьи и ее статус</p> <p><b>Фактический результат:</b> Происходит обновление страницы, появляется форма статьи и ее статус</p>
Название	Загрузка статьи с заполнением формы	
Приоритет	Высокий	
Устройство	Омега 68.0	
Версия приложения	03.06.2020	
Серьезность	Критическая	
Дополнения		

Рисунок 28 - Участник. Загрузка статьи



Идентификатор	red-01	<b>Предусловие:</b> Пользователь авторизован Пользователь имеет роль Редактора Пользователь находится в Моей Профиле Пользователь нажал на кнопку Статьи пользователей <b>Шаги:</b> 1. Выбрать пользователя uchastnik_test для смены роли на Редактора 2. Нажать на выпадающее меню 3. Выбрать пункт Принята 4. Нажать на кнопку Отправить <b>Ожидаемый результат:</b> Происходит перезагрузка страницы, статус статьи у uchastnik_test - Принята <b>Фактический результат:</b> Происходит падение после нажатия на кнопку Отправить
Название	Изменение состояния статьи Участника	
Приоритет	Высокий	
Устройство	Орега 68.0	
Версия приложения	03.06.2020	
Серьезность	Блокирующая	
Дополнения		

Рисунок 30 - Редактор. Изменение состояния статьи Участника

Идентификатор	ad-02	<b>Предусловие:</b> Пользователь авторизован Пользователь имеет роль Администратора Пользователь находится в Моей Профиле Пользователь нажал на кнопку Список пользователей <b>Шаги:</b> 1. Выбрать пользователя redactor_test для смены роли на Редактора 2. Нажать на выпадающее меню 3. Выбрать роль Редактор 4. Нажать на кнопку Отправить <b>Ожидаемый результат:</b> Происходит перезагрузка страницы, redactor_test имеет роль Редактора <b>Фактический результат:</b> Происходит падение после нажатия на кнопку Отправить
Название	Изменение роли Пользователей	
Приоритет	Высокий	
Устройство	Орега 68.0	
Версия приложения	03.06.2020	
Серьезность	Блокирующая	
Дополнения		

Рисунок 29 - Администратор. Изменение роли пользователей (зарегистрированных участников)

## 16. Реализация

Гость при переходе по ссылке на сайт попадает на Главную страницу сайта.

Гость, как и любой другой пользователь, может переключаться между страницами.

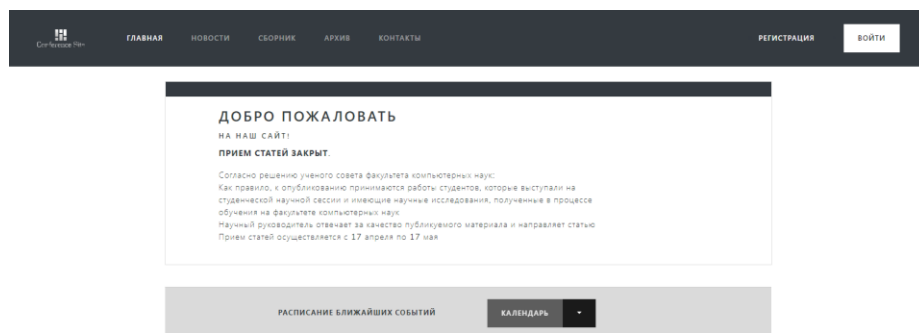


Рисунок 31 - Главная страница сайта

Администратор может выкладывать новости, относящиеся к конференции

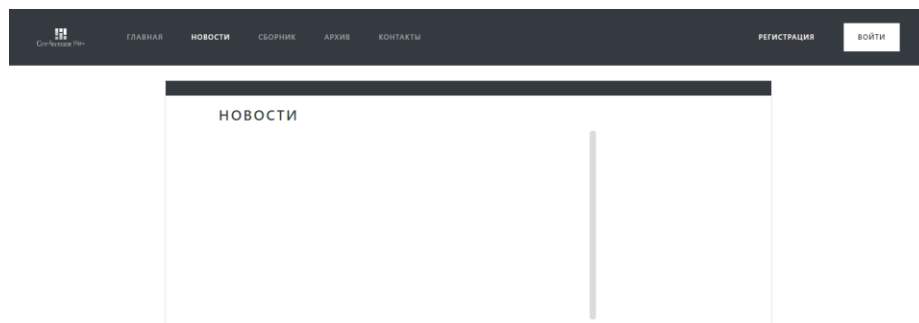


Рисунок 32 - Новости

На странице Сборники, любой пользователь может ознакомиться с основными правилами подачи работ. Также есть возможность скачать Шаблон и Пример в форматах .doc и Архив примеров в формате .rar



Рисунок 33 - Страница Сборник

На странице Архив все пользователи могут скачать Архивы работ за прошлые года

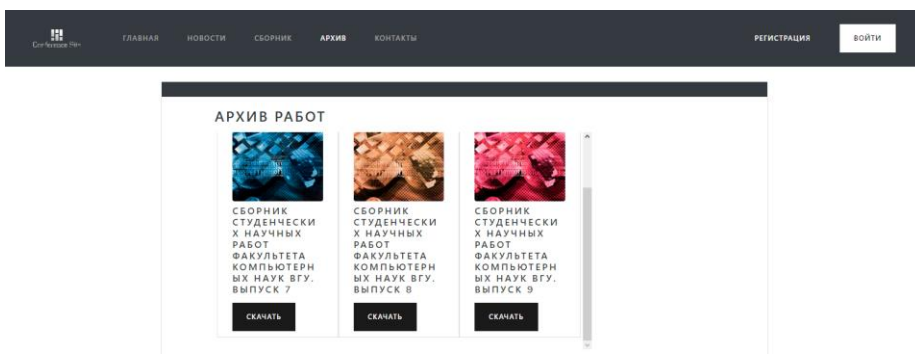


Рисунок 34 - страница Архив

На странице Контакты можно ознакомиться с контактами и рассмотреть карту местоположения Конференции.

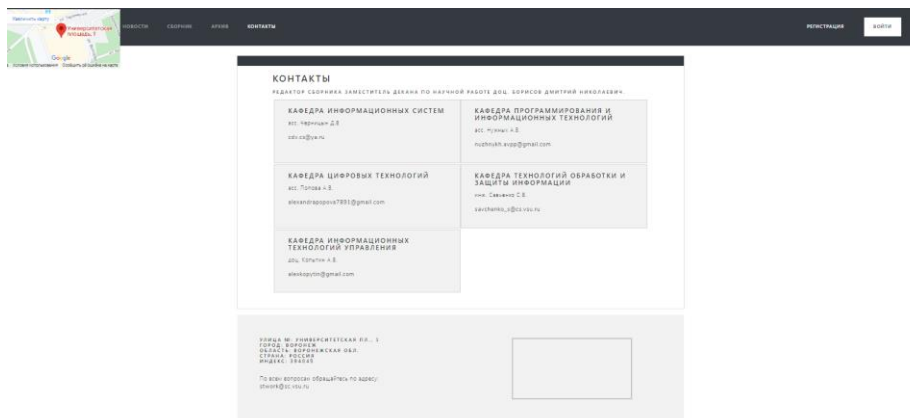


Рисунок 35 - страница Контакты

На странице Регистрации есть возможность зарегистрироваться, причем, пароль должен быть больше 6 символов и email должен email определенного формата (иначе появиться оповещение).

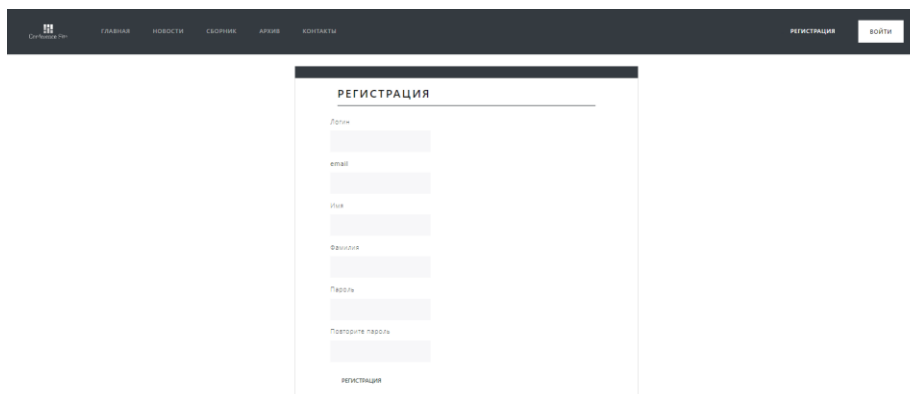


Рисунок 36 - страница Регистрации

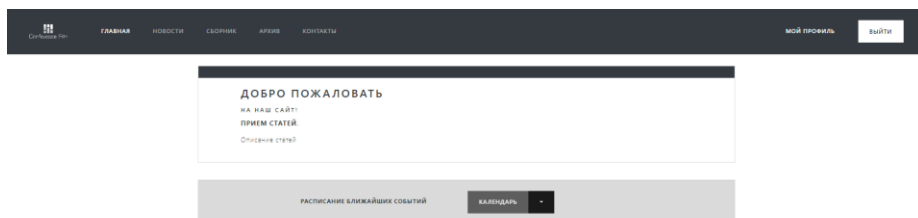


Рисунок 37 - страница Авторизации

В профиле Участника можно видеть диалоги Участника с другими зарегистрированными пользователями, так же Участник может регистрировать свой профиль.

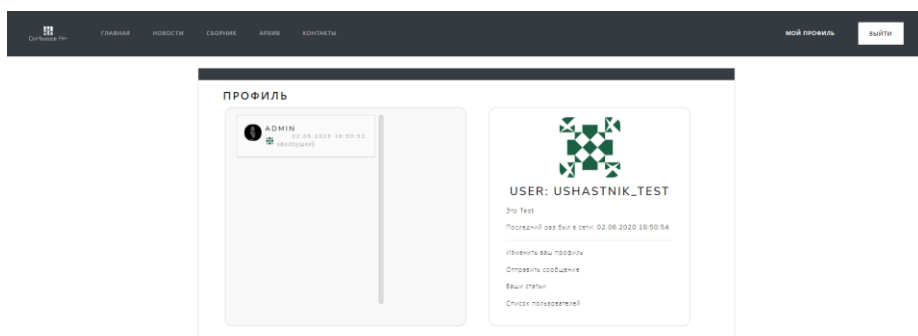


Рисунок 38 -Профиль Зарегистрированного пользователя + чат

Профиль Администратора похож на профиль Участника, но имеет дополнительные функциональные возможности: Статьи Пользователя и Список Пользователей.

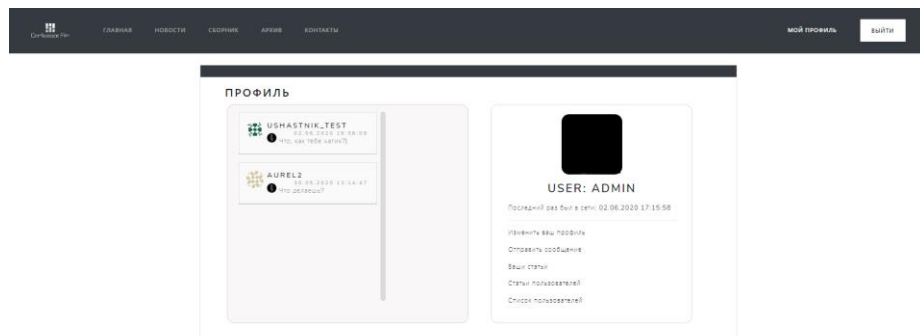


Рисунок 39 - Профиль Администратора

Администратор может просматривать статьи других участников и даже редакторов.

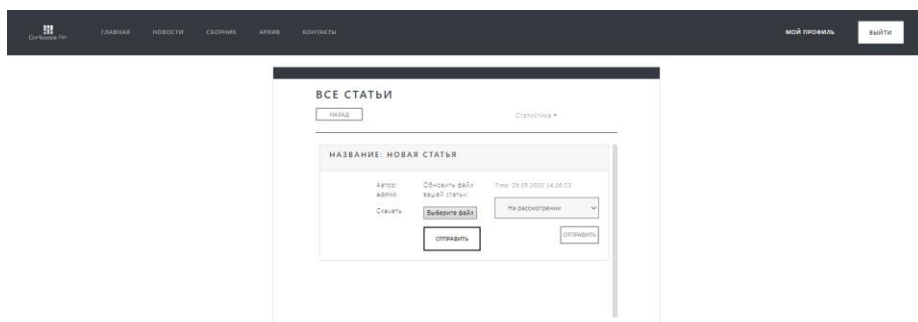


Рисунок 40 - Просмотр статей Администратором

Администратор может добавить любую Новость на сайт.

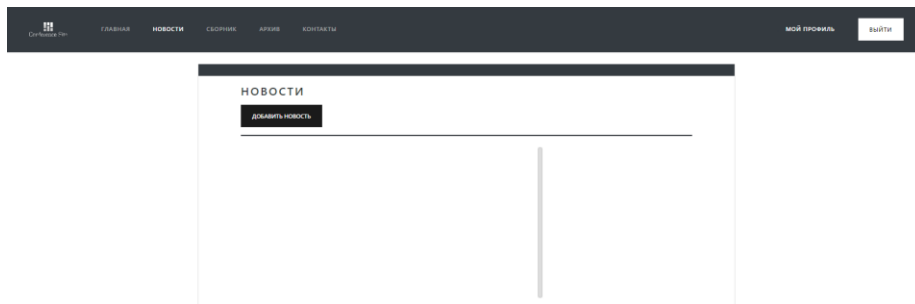


Рисунок 41 - Добавление Новостей на сайт

Администратор имеет полномочия для изменения роли любого пользователя по его просьбе.

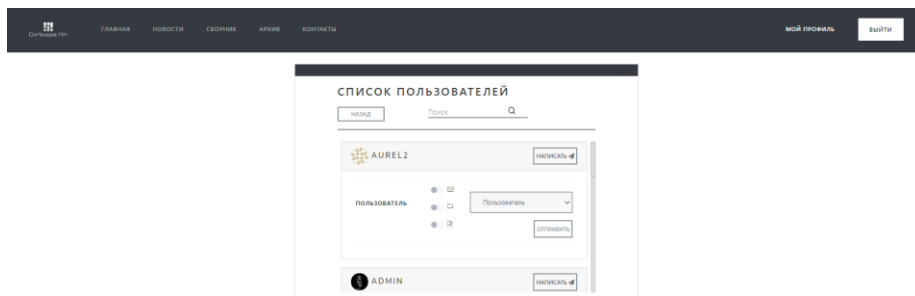


Рисунок 42 - Список Пользователей Администратора

У Редактора функциональные возможности шире, чем у Участника, но уже чем у Редактора. Редактор может скачивать статьи и выставлять им статус, в данном случае, статус – Принята. Так же он хранит всю статистику о всех работах (такую же функцию имеет и Администратор)

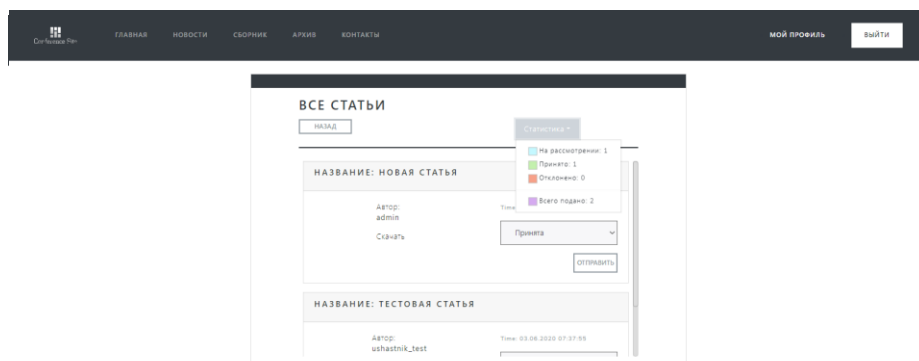


Рисунок 43 - статьи, которые может просмотреть Редактор



## **17. Заключение**

В ходе данной курсовой работы была произведена разработка Сервиса по подаче научных статей на Конференцию.

На основе аналитических данных воронок, можно сделать вывод, что поставленные задачи курсового проекта выполнены. Сервис отвечает заявленным функциональным возможностям и имеет широкие возможности по дальнейшему расширению.

Серверная часть приложения была разработана на языке Python с использованием фреймворка flask и Jinja2. Клиентская часть написана на языках HTML и CSS с использованием JQuery и фреймворка Bootstrap. Информация о пользователях и статьях хранится в базе данных SQLite.

Перед разработкой были произведены:

1. определение технических требований к web-сервису
2. проектирование web-сервиса средствами языка UML

После разработки было произведено тестирование и запись демо-видео. Приложение отвечает заявленным в Техническом Задании функциональным возможностям

## 18. Приложение

— Яндекс.Метрика/[сайт]:

URL:<https://metrika.yandex.ru/dashboard?group=day&period=week&id=62062741>

— Тест-кейсы/[сайт]:

URL:<https://docs.google.com/spreadsheets/d/15ziRzp6GtPHFU8mAYSu6ar6Az0sF-Q8FW5QEgxNAxfQ/edit#gid=1048512946>

## 19. Список использованных источников

1. Python Documentation / [сайт]. – URL: <https://www.python.org/doc/>
2. НОУ Интуит / [сайт]. – URL: <https://www.intuit.ru/studies/courses/1007/229/lecture/5950>
3. Леоненков. Самоучитель по UML / [сайт]. – URL: <http://khp-iip.mipk.kharkiv.edu/library/case/leon/index.html>
4. SQLAlchemy / [сайт]. – URL: <https://www.sqlalchemy.org>
5. Bootsnipp / [сайт]. – URL: <https://bootsnipp.com>
6. Bootswatch / [сайт]. – URL: <https://bootswatch.com>
7. Namecheap / [сайт]. – URL: <https://www.namecheap.com/logo-maker/app/>
8. HTML CSS JavaScript MySQL PHP Bootstrap book academy примеры онлайн / [сайт]. – URL: <https://html5css.ru>
9. Хабр: Паттерны для новичков: MVC vs MVP vs MVVM / [сайт]. – URL: <https://habr.com/ru/post/215605/>
10. Хабр: Мера-Учебник Flask Глава 1: Привет, мир! ( издание 2018 ) / [сайт]. – URL: <https://habr.com/ru/post/346306/>
11. Flask: User Guide <https://flask.palletsprojects.com/en/1.1.x/> / [сайт]. – URL: <https://flask.palletsprojects.com/en/1.1.x/>
12. Quora: What is the difference between an MVC and an MVT framework? / [сайт]. – URL: <https://www.quora.com/What-is-the-difference-between-an-MVC-and-an-MVT-framework>
13. Real Python: Model-View-Controller (MVC) Explained – With Legos ? / [сайт]. – URL: <https://realpython.com/the-model-view-controller-mvc-paradigm-summarized-with-legos/>